

General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

University of California, Los Angeles

FINAL REPORT

DATAFLOW COMPUTING APPROACH IN HIGH-SPEED DIGITAL SIMULATION

NASA Grant NAG 3-132

December 15, 1980 - December 14, 1982

National Aeronautics and Space Administration - Lewis Research Center

Principal Investigator: Milos D. Ercegovac, Associate Professor

Co-Principal Investigator: Walter J. Karplus, Professor

The overall objective of the above research program was to explore new computational tools and methodologies for the digital simulation of continuous systems [Karp 82]. The research was directed towards achieving programmability and cost-effective performance in multiprocessor organizations for real-time simulation. Our approach, based on functional-style languages and dataflow computing principles, which allow for the natural representation of parallelism in algorithms and provides a suitable basis for the design of cost-effective, high-performance distributed systems. Specifically, the objectives of this research were to (a) perform comparative evaluation of several existing dataflow languages and develop an experimental dataflow language suitable for real-time simulation using multiprocessor systems; (b) investigate the main issues that arise in the architecture and organization of dataflow multiprocessors for real-time simulation [Gaud 82a], and (c) develop and apply performance evaluation models in typical applications.

Summary of Results

The research in this period has included the following activities:

(a) Evaluation of several dataflow languages such as MIT VAL, UC Irvine Id, University of Manchester SIMPLE, and CERT LAU in the context of real time simulation problems. Specifically, we are interested in their capabilities and limitations with respect to (i) specification of simulation problems, (ii) simplicity and power in representing concurrency, and (iii) suitability for multiprocessor organizations. The evaluation criteria are based on several non-procedural languages presently used in simulation, such

(NASA-CR-173552) DATAFLOW COMPUTING
APPROACH IN HIGH-SPEED DIGITAL SIMULATION
Final Report, 15 Dec. 1980 - 14 Dec. 1982
(California Univ.) 5 p HC A02/MF A01

N84-25336

Unclas

CSCI 09B G3/61 19328

as ACSL and CSSL-IV. A. Makoui, a Ph.D. candidate, investigated user-related language properties and programming in this context [Mako 84].

(b) Specification and development of an experimental dataflow language. Such a language should have a strong mathematical basis and a very simple semantics. In such a language one needs to define only a small number of primitives and have simple ways of combining them into powerful higher level functions. Initially, D. Lahti implemented as a M.S. thesis project a functional language (BFL) based on Backus' FP language [Lahti 81]. D. Patel, a Ph.D. student, implemented an enhanced version of BFL using the FP system developed at UC Berkeley by S. Baden. The system provides several automatic performance evaluation facilities. User can specify tracing of the number of steps and the amount of concurrency at each step for the selected functions in the program and obtain useful information about the performance of the algorithm. The partitioning and allocation in the case of limited resources is also being studied, in particular in the context of the BFL compiler. It should be stressed that programs written in a language without side-effects, such as BFL, lend themselves nicely to the analysis required for partitioning and allocation and this is one of the main reasons for their use in our research project. BFL allows programming in a highly mathematical style using data objects such as vectors and arrays. Briefly, the language consists of the set of objects, a set of primitive functions (such as arithmetic, logical and object manipulation functions), a set of composition forms (to allow construction of programs out of simpler programs) and a set of definitions (user-defined functions). As an initial effort in adapting BFL to our purposes, S.L. Lu has implemented a compiler for BFL which allows translation of functional programs into a conventional machine language code [Lu 84]. The compiler is written in such a way that the target object code can be changed without difficulties. A number of typical simulation problems has been programmed in order to gain some insight into the user interface requirements and characteristics of functional programming. A discussion of this approach and some experimental results are reported in [Erce 83].

(c) The problem of resolution (atomicity) in dataflow architectures and its effects on performance has been formulated and studied. Some results on the use of a higher-level resolution in handling array data structures were presented in [Gaud 82a, Gaud 82b]. Other implementation issues in dataflow systems, such as the implementation of optimizing pipelines for programs expressed as dataflow graphs, have been investigated [Tong 81].

(d) An initial specification of a baseline multi-microprocessor for execution of functional programs compiled into conventional machine multi-task code is under development. A broadcast-type multiprocessor organization was proposed by M. Ercegovac and P.K. Chan began investigation on the basic hardware requirements and mechanisms for its implementation.

Copies of references [Erce 83], [Karp 82] and [Gaud82a] are included in the appendix of the final report.

Publications

- [Erce 83] Ercegovac, M.D., and Lu, S.L., "A Functional Language Approach in High-Speed Digital Simulation", Proc. Summer Computer Simulation Conference, Vancouver, B.C., 1983.
- [Gaud 82a] Gaudiot, J.L., and Ercegovac, M.D., "A Scheme for Handling Arrays in Data-Flow Systems", Proc. 3rd International Conference on Distributed Computing Systems, Miami, Florida, 1982.
- [Gaud 82b] Gaudiot, J.-L., "On Program Decomposition and Partitioning in Data-Flow Systems", Ph.D. Dissertation, UCLA Technical Report No. CSD-8212212, December 1982.
- [Karp 82] Karplus, W.J. and Makoui, A., "The Role of Dataflow Methods in Continuous Systems Simulation", Proc. Summer Computer Simulation Conference, Denver, Colorado, 1982.
- [Lahti 81] Lahti, D.O., "Applications of a Functional Programming Language", MS Thesis, UCLA Computer Science Department, Report No. CSD-810403, April 1981.
- [Lu 82], Lu, S.L., "A Functional Programming System - Its Compiler and System Architecture", MS Thesis, UCLA Computer Science Department, 1984.
- [Mako 84] Makoui, A., "(his dissertation title)", Ph.D. Dissertation, UCLA Computer Science Department, 1984.
- [Tong 81] Tong, S.F., "Implementation of Optimizing Pipelines for Data Flow Programs", MS Thesis, UCLA Computer Science Department, 1981.

Personnel 1980-1982

Graduate Students

Chan, P.K.

Gaudiot, J.-L.

Lu, S.L.

Makui, A.

Tong, S.F.

Staff

Myers, J.

Appendix

.....